Jurnal Riset Mahasiswa Matematika
Volume 2(3) (2023), Pages 84-89
p-ISSN: 2808-1552; e-ISSN: 2808-4926

Membangun Super Enkripsi untuk Mengamankan Pesan

Laura Agustina*, Imam Sujarwo, Muhammad Khudzaifah

Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia

lauraagustina48@gmail.com*, imsuha@mat.uin-malang.ac.id, khudzaifah@uin-malang.ac.id

Abstrak

Masalah keamanan pesan atau suatu informasi adalah masalah sangat penting. Makna keamanan di sini mengacu pada suatu pesan yang bersifat rahasia yang akan disampaikan kepada penerima. Sebuah ilmu yang mempelajari tentang pengamanan kerahasiaan pesan/data dengan menggunakan sandi disebut kriptografi. Untuk lebih meningkatkan keamanan maka dilakukan penggabungan dua algoritma untuk mengamankan suatu pesan. Super enkripsi merupakan suatu konsep yang menggunakan kombinasi dari dua atau lebih teknik kriptografi substitusi dan permutasi (transposisi) untuk mendapatkan suatu algoritma yang lebih sulit untuk dipecahkan. Hal pertama yang dilakukan adalah melakukan enkripsi pesan dengan menggunakan teknik substitusi (*Cipher* Substitusi), kemudian dienkripsi lagi dengan menggunakan teknik permutasi (*Cipher* Transposisi). Pada penelitian kali ini akan dilakukan penggabungan dua algoritma kriptografi untuk membangun super enkripsi menggunakan algoritma *Vigenere Cipher* dan *Bifid Cipher* dalam mengamankan pesan. Proses enkripsi pesan yaitu menggunakan algoritma *Vigenere Cipher* untuk proses enkripsi yang pertama, kemudian dilanjutkan menggunakan algoritma *Bifid Cipher* untuk proses enkripsi yang kedua. Untuk proses dekripsi dilakukan sebaliknya, dimulai dari urutan terbelakang proses enkripsi. Penggabungan dua algoritma ini menghasilkan keamanan pesan yang lebih terjaga.

Kata kunci: *Bifid Cipher*; Dekripsi; Enkripsi; Super Enkripsi; *Vigenere Cipher*

Abstract

The issue of message security or an information is very important. A science that studies about securing the confidentiality of messages using passwords is called cryptography. To enhance security, two algorithms are combined to secure messages. Super encryption is a concept that uses a combination of two or more substitution and permutation (transposition) cryptography techniques to obtain an algorithm that is more difficult to crack. The first thing to do is to encrypt the message using a substitution technique (Cipher Substitution), then re-encrypt it using a permutation technique (Cipher Transposition). In this study, two cryptographic algorithms will be combined to build super encryption using the Vigenere Cipher and Bifid Cipher algorithms to secure messages. The message encryption process is using the Vigenere Cipher algorithm for the first encryption process, then continued using the Bifid Cipher algorithm for the second encryption process. The encryption process is done the other way around, starting from the back of the encryption process. The combination of these two algorithms results in more secure message security.

Keywords: Bifid Cipher; Decryption; Encryption; Super Encryption; Vigenere Cipher

PENDAHULUAN

Pesan adalah sebuah pemberitahuan, kata, perintah, informasi, atau komunikasi dalam bentuk lisan maupun tulisan yang dibuat oleh pengirim kepada penerima pesan. Proses pengiriman pesan tentu dibutuhkan suatu media atau perantara agar pesan dapat tersampaikan dan diterima dengan baik oleh penerima pesan. Masalah keamanan pesan atau suatu informasi sangat penting. Makna keamanan di sini adalah suatu pesan bersifat rahasia yang akan disampaikan kepada penerima. Tidak ada pihak-pihak lain yang boleh mengetahui isi informasi ataupun pesan kecuali yang memiliki kewenangan, seperti pengirim dan penerima pesan.

Submitted: 6 Juli 2022 Reviewed: 21 Desember 2022 Accepted: 28 Februari 2023
Doi: http://dx.doi.org/10.18860/jrmm.v2i3.16335

Sebuah ilmu yang mempelajari tentang pengamanan kerahasiaan pesan/data dengan menggunakan sandi disebut kriptografi. Sekarang kriptografi dapat diartikan sebagai ilmu yang berdasar pada teknik perhitungan matematika untuk keamanan suatu informasi [1]. Kriptografi digunakan untuk menjaga pesan rahasia dengan cara mengubah ke dalam susunan pesan yang tidak dimengerti menggunakan sebuah sandi sehingga pesan tersebut terlihat seperti susunan huruf acak. Fungsi kriptografi bukan hanya sekedar kerahasiaan data *(privacy*) saja, tetapi juga bertujuan untuk menjaga integritas data (*data integrity*), keaslian data (*authentication*) dan anti penyangkalan (*nonrepudation*) [2] [3].

Enkripsi bertujuan untuk menjaga informasi rahasia/tersembunyi dari seseorang yang tidak berhak mengetahui informasi rahasia tersebut, serta menjamin kerahasiaan terhadap informasi yang telah dienkripsi. Sedangkan dekripsi merupakan kebalikan dari proses enkripsi, yaitu transformasi dari informasi rahasia yang telah melalui proses enkripsi yang menghasilkan *ciphertext* (pesan rahasia) dan kembali ke bentuk semula *plaintext* (pesan asli) [4]. Algoritma enkripsi merupakan fungsi yang digunakan untuk melakukan proses enkripsi dan dekripsi. Algoritma ini dibuktikan menggunakan basis matematika [5].

Salah satu cara agar pesan menjadi sulit dipecahkan yaitu dengan mengomposisikan dua *cipher* [6]. Untuk lebih meningkatkan keamanan maka dilakukan penggabungan dua algoritma untuk mengamankan suatu pesan. Super enkripsi yang merupakan suatu konsep kombinasi dari dua atau lebih algoritma kriptografi yaitu teknik substitusi dan permutasi *cipher* untuk mendapatkan suatu algoritma yang sulit dipecahkan [7]. Hal pertama yang harus dilakukan adalah menggunakan teknik substitusi (*Cipher* Substitusi) untuk mengenkripsi pesan, dan kemudian menggunakan teknik permutasi (*Cipher* Transposisi) untuk mengenkripsi kembali *ciphertext* yang diperoleh [8]. Untuk proses dekripsi dilakukan sebaliknya, dimulai dari urutan terbelakang proses enkripsi.

Prinsip utama teknik substitusi adalah mengganti kemunculan sebuah karakter dengan karakter lainnya. Sistem kriptografi yang menggunakan operasi substitusi disebut dengan sistem kriptografi berbasis substitusi [9]. Sedangkan prinsip teknik transposisi berlawanan dengan teknik substitusi, yaitu dengan mengubah posisi karakter untuk mendapatkan *ciphertext* tanpa mengganti dengan karakter lain. Sistem ini disebut dengan sistem kriptografi berbasis transposisi/permutasi [3] [10].

Sandi *Vigenere* merupakan algoritma substitusi yang diperoleh dari modifikasi Sandi *Caesar*. Teknik ini diberi nama sesuai dengan penemunya, yaitu *Blaise de Vigenere* dari Istana Henry III Prancis pada abad ke-16, dan dianggap tidak dapat dipecahkan selama 300 tahun. Sandi *Vigenere* merupakan metode enkripsi huruf dengan menggunakan serangkaian enkripsi Sandi Caesar, tetapi kunci yang digunakan pada Sandi *Vigenere* berupa huruf yang disusun dan membentuk suatu kata [11]. Dengan cara ini, setiap huruf dari *plaintext* yang sama dapat memiliki huruf *ciphertext* yang berbeda, tergantung pada kunci huruf yang digunakan. Jadi dengan menggunakan Vigenere *password*, kita dapat mencegah frekuensi huruf pada *ciphertext* muncul sama dengan frekuensi huruf pada *plaintext* [12]. Sama seperti *Caesar Cipher*, untuk penomoran karakter huruf yaitu $0-25$ dengan $A=0, B=1, \dots, Z=25$, dan penjumlahan dilakukan dengan *modulo* 26 karena karakter huruf hanya berhenti hingga angka 25. Enkripsi Sandi *Vigenere* teks pesan asli (*plaintext*) dengan kunci k dapat dijelaskan secara matematis sebagai [11]:

$$E: C_i = (P_i + K_i)\ mod\ 26 \quad (1)$$

dekripsi dilakukan dengan cara yang sama, yaitu:

$$D: P_i = (C_i - K_i)\ mod\ 26 \quad (2)$$

di mana

E : Proses Enkripsi
D : Proses Dekripsi
P_i : Karakter Plaintext ke-i
C_i : Karakter Ciphertext ke-i
K_i : Karakter Kunci ke-i
i : urutan karakter ke-i

Bifid Cipher adalah berbentuk matriks, atau termasuk ke dalam *cipher* transposisi. *Bifid Cipher* berbentuk matriks dengan ordo 5×5. Algoritma *Bifid Cipher* tergolong pada algoritma

kriptografi klasik yang ditemukan oleh Felix Delestelle. Algoritma ini menggunakan persegi *polibius* dan pemecahan karakter untuk memperoleh efek difusi pada *ciphertext*-nya [2]. Sampai saat ini *Bifid Cipher* masih merupakan algoritma penyandian sederhana, biasanya hanya menggunakan pensil dan kertas yang sangat aman digunakan untuk menyelesaikan. *Bifid Cipher* adalah algoritma yang menggunakan metode *playfair cipher*, di mana teks pesan asli (*plaintext*) diacak menjadi tabel persegi 5×5 dengan mengacak posisi baris dan kolom huruf pada pesan yang sesuai dengan papan kunci (bentuk matriks), seperti penerapan pada *playfair cipher*. Setiap baris dan kolom pada susunan papan kunci diisi dengan huruf alfabet yang disusun secara acak [13]. Tabel bawaan yang diperkenalkan oleh *playfair* adalah tabel matriks 5×5 yang berisi huruf kapital $A - Z$, dengan huruf J dihilangkan dan diganti dengan huruf I. Berikut adalah beberapa aturan yang digunakan dalam metode sandi *Playfair* [14]:

1. Jika ada huruf J, ganti dengan huruf I. Dalam beberapa versi, yang dihilangkan adalah huruf Q, sedang dalam versi lain huruf I dan J ditulis sebagai I / J [12].
2. Tulis pesan dalam pasangan huruf, yaitu memisahkan dua huruf.
3. Tidak boleh ada huruf yang sama, jadi sisipkan huruf atau karakter yang jarang digunakan di tengah huruf yang sama (ganda).
4. Jika jumlah huruf ganjil, tambahkan huruf atau karakter yang jarang digunakan di akhir dari tabel dibentuk.

METODE

Adapun langkah-langkah yang harus dilakukan untuk menyelesaikan super enkripsi tersebut adalah:

Proses Enkripsi

1. Membuat *Plaintext*.
2. Menentukan kunci.
3. Melakukan enkripsi menggunakan algoritma *Vigenere Cipher*.
4. Memperoleh *Ciphertext* dari hasil enkripsi menggunakan algoritma *Vigenere Cipher*.
5. *Ciphertext* tersebut selanjutnya menjadi *Plaintext* untuk algoritma *Bifid Cipher*.
6. Melakukan enkripsi menggunakan algoritma *Bifid Cipher*.
7. Memperoleh *Ciphertext* akhir dari hasil enkripsi menggunakan gabungan algoritma *Vigenere Cipher* dan *Bifid Cipher*.

Proses Dekripsi

1. Menentukan *Ciphertext*.
2. Mendapatkan kunci.
3. Melakukan dekripsi menggunakan algoritma *Bifid Cipher*.
4. Memperoleh *Plaintext* dari hasil dekripsi menggunakan algoritma *Bifid Cipher*.
5. *Plaintext* tersebut selanjutnya menjadi *Ciphertext* untuk algoritma *Vigenere Cipher*.
6. Melakukan dekripsi menggunakan algoritma *Vigenere Cipher*.
7. Memperoleh *Plaintext* akhir dari hasil dekripsi menggunakan gabungan algoritma *Vigenere Cipher* dan *Bifid Cipher*.

HASIL DAN PEMBAHASAN

Untuk implementasi proses enkripsi dan dekripsi dapat dilakukan dengan pemrograman *Python*. Karena *Python* merupakan bahasa pemrograman yang mudah dipelajari serta memiliki kode-kode pemrograman yang lengkap, jelas, dan mudah dipahami. *Python* dapat digunakan untuk pembuatan aplikasi berbasis kecerdasan buatan (*artificial intelligence*). Distribusi aplikasi yang dibuat menggunakan *Python* bersifat *multi-platform*, yang artinya dapat digunakan dan dijalankan pada berbagai platform sistem operasi [15].

Pada penelitian ini akan menjelaskan algoritma matematika untuk implementasi enkripsi dan dekripsi sebelum menggunakan bahasa pemrograman *Python.* Urutan algoritma yang digunakan pada proses enkripsi adalah algoritma *Vigenere Cipher* kemudian dilanjutkan dengan algoritma *Bifid Cipher*. Dan urutan algoritma yang digunakan pada proses dekripsi adalah kebalikan dari proses enkripsi yaitu algoritma *Bifid Cipher* kemudian dilanjutkan dengan algoritma *Vigenere Cipher*.

Untuk algoritma *Bifid Cipher* di sini akan dilakukan proses enkripsi dan dekripsi menggunakan papan kunci berukuran 6×6, hal ini bertujuan untuk mempermudah dalam pembuatan program enkripsi dan dekripsi menggunakan *Python*. Dengan papan kunci berukuran 6×6, penulisan huruf I dan J pada tabel dapat diletakkan pada posisi yang berbeda tempat. Sebagai pelengkap agar dapat menjadi papan kunci berbentuk persegi, dilakukan penambahan angka 0 sampai dengan 9. Digunakan algoritma ini karena kedua algoritma tersebut memiliki karakter kunci yang sama.

Berikut akan dituliskan algoritma matematika/*pseudocode* untuk algoritma super enkripsi:

Proses Enkripsi

Input : Plaintext(P), key(k)
Output : Ciphertext Akhir
Deklarasi :

```
Key, newKey, Plaintext : string;
Karakter : array[1...karakter.length]of string;
i, j : integer;
```

Proses :

```
input (Plaintext)
input (key);
for (int i ← 0; i < Plaintext.length)do
      j ← i mod key.length();
      newKey ← key(j);
end for;
      write(newKey);
      karakter [i] ← Plaintext;
      key ← get(key. Plaintext);
for (int i ← 0; i < karakter.length)do
      Ciphertext ← (char) (((26+((karakter[i]-‘ ‘)+key(i) – ‘ ‘))mod 26+‘
‘);
end for;
write (Ciphertext1);

Plaintext ← Ciphertext1
Alphabet ← “ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789”
Declare Integer new_places[Plaintext.length * 2 - 1]
position ← 0
for i ← 0 to Plaintext.length - 1
      position ← Alphabet.indexof(Plaintext(i))
      new_places [i] ← position DIV 6 + 1
      new_places [i + Plaintext.length] ← position MOD 6 + 1
end for
Ciphertext Akhir ← “ “
counter ← 0
for i ← 0 to Plaintext.length – 1
       Ciphertext Akhir ← Ciphertext Akhir +
```

```
                    Alphabet(((new_places[counter] – 1 * 5) +
                    new_places[counter + 1] – 1)
                counter ← counter + 2
        end for
        output Ciphertext Akhir
```

Proses Dekripsi

Input : Ciphertext Akhir(C), key(k)
Output : Plaintext Akhir(P)
Deklarasi :

```
        Key, newKey, Ciphertext Akhir : string;
        Karakter : array[1...karakter.length]of string;
        i, j : integer;
```

Proses :

```
        Alphabet ← ‘‘ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789‘‘
        Ciphertext ← Ciphertext Akhir n Ciphertext
        Declare Integer new_places[Plaintext.length * 2 - 1]
        counter ← 0
        for i ← 0 to Ciphertext.length – 1
              position ← Alphabet.indexof(Plaintext(i))
              new_places [counter] ← position DIV 6 + 1
              new_places [counter + 1] ← position MOD 6 + 1
              counter ← counter + 2
        end for
        Plaintext1 ← “ “
        for i ← 0 to Ciphertext.length – 1
              Plaintext1 ← Alphabet(((new_places[counter] – 1) * 5) +
                          new_places[counter + Ciphertext.length] -1)
        end for
        output Plaintext1

        Ciphertext ← Plaintext1
        for (int i ← 0; i < Ciphertext.length)do
              j ← i mod key.length();
              newKey ← key.charAt(j);
        end for;
              write(newKey);
              karakter[i] ← Ciphertext;
              key ← get(key. Ciphertext);
        for (int i ← 0; i < karakter.length)do
              Plaintext Akhir ← (char) (((26+((karakter[i] – ‘ ‘) –
                                key.charAt(i) – ‘ ‘)) mod 26) + ‘ ‘);
        end for
        write (Plaintext Akhir);
```

KESIMPULAN

Implementasi algoritma super enkripsi menggunakan pemrograman *Python* dapat dilakukan dengan menyusun proses penghitungan secara matematis ke dalam bentuk susunan algoritma matematika / *pseudocode,* kemudian diubah ke dalam bahasa pemrograman *Python* sesuai dengan prosedur dan langkah-langkah proses enkripsi dan dekripsi pesan. Keunggulan dari super enkripsi menggunakan algoritma *Vigenere Cipher* dan algoritma *Bifid Cipher* adalah hanya

menggunakan satu kunci saja, karena meskipun jenis kedua algoritma tersebut berbeda, tetapi memiliki karakter kunci yang sama yaitu menggunakan karakter kata.

DAFTAR PUSTAKA

[1] H. Sahara, "Implementasi Pengamanan Pesan Chatting Menggunakan Metode Vigenere Cipher dan Cipher Block Chainning," *Media Informasi Analisa dan Sistem(MEANS),* vol. 3, pp. 166-193, 2018.

[2] D. I. Manullang, "Perancangan Aplikasi Penyandian File Teks Dengan Algoritma Bifid Cipher," *Jurnal Pelita Informatika,* vol. 6, pp. 319-324, 2018.

[3] R. A. M. d. F. A. Rafrastara, "Super Enkripsi Teks Kriptografi Menggunakan Algoritma Hill Cipher dan Transposisi Kolom," *Prosiding SENDI_U ,* pp. 85-92, 2019.

[4] R. K. P. P. d. F. Latifah, "Implementasi Enkripsi Dekripsi Pesan Teks Menggunakan Model Julius Caesar Berbasis Object Oriented Programme," *Jurnal Techno Nusa Mandiri,* vol. 11, pp. 17-26, 2014.

[5] S. Aprilia, "Pengamanan Data Informasi Menggunakan Kriptografi Klasik," *Jurnal Teknik Informatika UIN Sunan Gunung Djati,* vol. 10, pp. 160-167, 2005.

[6] K. Y. d. H. S. H. Muhammad Lutfi Wijaya, "Kriptografi dengan Komposisi Caesar Cipher dan Affine Cipher untuk Mengubah Pesan Rahasia," *EurekaMatika,* vol. 5, pp. 30-45, 2017.

[7] D. Ariyus, Kriptografi Keamanan Data dan Komunikasi, Yogyakarta: Graha Ilmu, 2006.

[8] D. Arius, Pengantar Ilmu Kriptografi Teori Analisis dan Implementasi, Yogyakarta: C.V Andi Offset, 2008.

[9] R. Sadikin, Kriptografi untuk Keamanan Jaringan, Yogyakarta: C.V Andi Offset, 2012.

[10] M. A. M. d. N. P. Sastra, "Efektivitas Pesan Teks dengan Cipher Substitusi, Vigenere Cipher, dan Cipher Transposisi," *Majalah Ilmiah Teknologi Elektro,* vol. 17, pp. 59-65, 2018.

[11] A. K. Quist, "A Hybrid Cryptosystem Based on Vigenere Cipher and Columnar Transposition Cipher," *International Journal of Advanced Tecnology & Engineering Research (IJATER),* vol. 3, no. 1, pp. 141-147, 2013.

[12] R. Munir, Kriptografi, Bandung: Informatika Bandung, 2019.

[13] S. Wulandari, "Pengamanan Pesan Teks E-Mail Menggunakan Metode Algoritma Bifid dan Feedback Cipher," *Jurnal Riset Komputer (JURIKOM),* vol. 6, pp. 523-530, 2019.

[14] D. Susanti, "Analisis Modifikasi Metode Playfair Cipher Dalam Pengamanan Data Teks," *Indonesian Journal of Data and Science,* vol. 1, pp. 11-18, 2020.

[15] J. Enterprise, Python untuk Programmer Pemula, Jakarta: PT. Elex Media Komputindo, 2019.